

Laporan Tugas 1
IF3260 Grafika Komputer
2D Web Based CAD (Computer-Aided Design)



Oleh :

13519168	Nabil Nabighah
13519187	Ahmad Saladin
13519205	Muhammad Rifat Abiwardani

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2022

I. Deskripsi

Dalam tugas ini, mahasiswa membuat program yang dapat memodelkan denah arsitektur suatu bangunan menggunakan empat model dasar. Denah digambar menggunakan garis, persegi, persegi panjang, dan poligon, dimana model disimpan dalam sebuah file yang dengan mudah diedit berisi daftar koordinat dan warna setiap polygon. Pengguna juga dapat membuka sebuah file model hasil penyimpanan. Program memiliki fitur interaktif berupa menggeser titik kontrol/simpul dengan mouse, mengubah panjang garis, mengubah ukuran sisi persegi, mengubah warna poligon, dan membuka menu help yang memudahkan pengguna baru untuk dapat melakukan operasi di atas tanpa harus bertanya. Program dikembangkan dengan WebGL murni tanpa menggunakan library atau framework tambahan.

II. Hasil

Kami membuat program berupa 2D Web Based CAD. Program menggunakan WebGL, HTML, CSS dan Javascript dalam pembuatannya. Link program dapat dilihat pada link berikut.

- <https://github.com/Uyamikun/tubes1-grafkom>

Program yang telah dibuat memanfaatkan fungsi-fungsi berikut.

No	Nama Fungsi	Deskripsi
1.	initCanvas, initGL, getShader, initProgram	Fungsi initCanvas mengambil DOM canvas jika ada pada halaman, kemudian initGL membuat objek WebGLRenderingContext yang digunakan untuk berinteraksi dengan GPU. Interaksi dicapai dengan mengirim shader yang diinisialisasi menggunakan getShader dan initProgram untuk melakukan linking awal.
2.	createPointsShaderProgram	Fungsi createPointsShaderProgram membuat objek WebGL program dengan meng-compile vertex shader dan fragment shader yang sesuai untuk objek titik, yakni

		vertex shader dengan atribut vektor posisi berukuran dua, pengaturan ukuran titik (dalam program ini 5), lalu men-set nilai GL position sebagai nilai vektor posisi, serta fragment shader yang nilai GL fragment color-nya konstan bernilai (0, 0, 0, 1) atau hitam.
3.	createLinesShaderProgram	Fungsi createPointsShaderProgram membuat objek WebGL program dengan meng-compile vertex shader dan fragment shader yang sesuai untuk objek titik, yakni vertex shader dengan atribut vektor posisi berukuran dua yang men-set nilai GL position sebagai nilai vektor posisi, serta fragment shader yang nilai GL fragment color-nya konstan bernilai (0, 0, 0, 1) atau hitam.
4.	createShapesShaderProgram	Fungsi createShapesShaderProgram membuat objek WebGL program dengan meng-compile vertex shader dan fragment shader yang sesuai untuk objek titik, yakni vertex shader dengan atribut vektor posisi berukuran dua dan atribut vektor warna berukuran empat, kemudian men-set nilai GL position sebagai nilai vektor posisi dan mengirim vektor warna ke fragment shader, dimana fragment shader men-set nilai GL fragment color sebagai nilai vektor warna tersebut.
5.	onCanvasMouseDown	Fungsi yang berfungsi untuk menangani ketika user menekan mouse dan memiliki penanganan berbeda sesuai dengan fungsionalitas yang pilih pengguna.
6.	onCanvasMouseMove	Fungsi yang berfungsi untuk menangani ketika user menggerakkan mouse setelah ditekan. Fungsi ini memiliki penanganan berbeda sesuai dengan fungsionalitas yang pilih pengguna.
7.	onCanvasMouseUp	Fungsi yang berfungsi untuk menangani ketika user selesai melepaskan mouse setelah ditekan. Fungsi ini memiliki penanganan berbeda sesuai dengan

		fungsi yang dipilih pengguna.
8.	download	Fungsi ini mengunduh sebuah file yang berisi hasil model yang telah dibuat.
9.	findDrawObjectPoint	Fungsi ini mengembalikan index dari objek dalam drawCollection beserta index dari titik dalam objek tersebut jika berada cukup dekat dengan suatu posisi input.
10.	findDrawObjectArea	Fungsi ini mengembalikan index dari objek dalam drawCollection ketika posisi input yang diberikan termasuk dalam area sebuah objek.
11.	convertHexToRGB	Fungsi mengubah input warna dari hexadecimal menjadi RGB.
12.	readColorPicker	Fungsi untuk membaca input warna yang dipilih pengguna.
13.	window.onload	Saat halaman dibuka, program diinisialisasi dengan pertama memanggil fungsi-fungsi utilitas initCanvas dan initGL, lalu menghapus canvas dan menghapus buffer bit, kemudian meng-attach fungsi onCanvasMouseDown, onCanvasMouseMove, dan onCanvasMouseUp pada event onmousedown, onmousemove, dan onmouseup dari objek DOM canvas. Terakhir, halaman diinisialisasi dengan mendefinisikan fungsi-fungsi onClick dari tombol-tombol yang ada pada halaman.
14.	render	Fungsi render pertama melakukan setup render time seperti resize canvas dan clear color buffer, lalu mengiterasi pada array drawCollection untuk menggambar masing-masing objek sesuai urutan kemunculan dalam array. Jika objek merupakan tipe garis, maka fungsi renderLine dan renderPoints dipanggil pada objek tersebut. Jika objek merupakan tipe persegi, persegi panjang, atau poligon, maka fungsi renderPolygon dipanggil pada objek tersebut, dengan kondisi dimana jika renderPolygon berhasil

		me-render objek, maka renderLine dan renderPoints dilakukan untuk objek tersebut. Setelah menangani semua objek pada array drawCollection, program menggambar poligon yang sedang digambar secara freehand oleh pengguna (jika ada) menggunakan fungsi renderDrawingPoligon.
15.	renderDrawingPoligon	Fungsi renderDrawingPoligon menerima input kumpulan titik terurut dan menggunakan shader program dari createLinesShaderProgram serta method drawArrays pada mode LINE_STRIP untuk menggambar sisi poligon yang sedang digambar.
16.	renderPolygon	Fungsi renderPolygon menerima input kumpulan titik terurut dari sebuah poligon, kemudian melakukan triangulasi agar dapat digambar menggunakan mode TRIANGLE dengan method drawArrays. Shader yang digunakan diperoleh dari fungsi createShapesShaderProgram.
17.	renderLine	Fungsi renderLine menerima input dua titik kemudian menggambar garis yang menghubungkan kedua titik tersebut menggunakan shader createLinesShaderProgram dan mode LINES pada method drawArrays.
18.	renderPoints	Fungsi renderPoints menerima input array titik kemudian menggambar setiap titik menggunakan shader createPointsShaderProgram dan mode POINTS pada method drawArrays.
19.	isPointOnSegment	Fungsi isPointOnSegment memeriksa apabila suatu titik berada pada segmen garis yang terdiri dari dua titik lain, dengan asumsi bahwa titik berada pada sepanjang ekstensi segmen garis, dengan cara memeriksa apakah letak titik berada di antara kedua titik pembangun segmen garis.

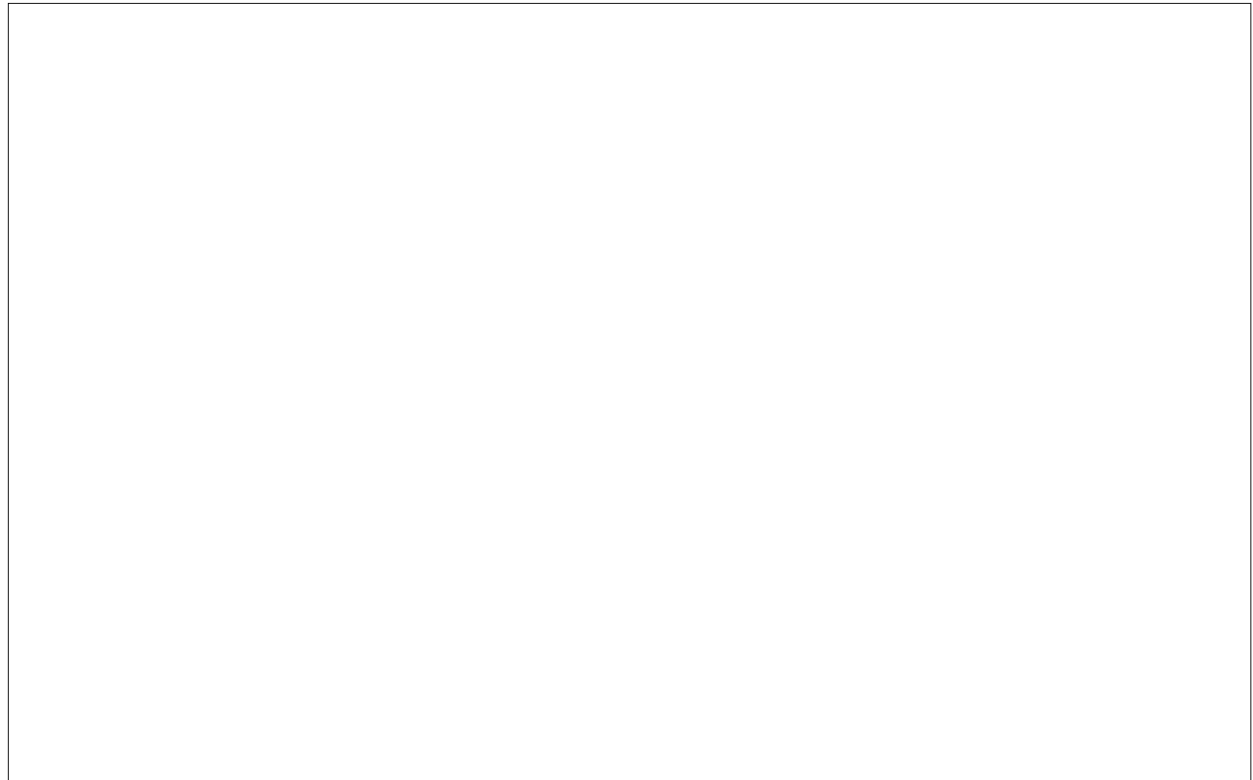
20.	cycleOrder	Fungsi cycleOrder menentukan apakah siklus tiga titik P1, P2, dan P3 searah jarum jam, berlawanan jarum jam, atau bukan keduanya (P1-P2-P3 berada pada garis yang sama) dengan menghitung nilai cross product.
21.	checkIntersect	Fungsi checkIntersect memeriksa apabila dua garis yang dibentuk oleh (P1, P2) dan (P3, P4) saling memotong. Algoritma menggunakan prinsip bahwa jika garis (P1, P2) memotong (P3, P4), maka urutan siklus P1-P2-P3 dan P1-P2-P4 harus berbeda, dan sebaliknya. Namun terdapat pengecualian dimana salah satu titik berada pada segmen garis lainnya, maka hanya diperiksa jika urutan siklus bukan searah jarum jam maupun berlawanan jarum jam.
22.	polygon_area	Fungsi polygon_area menghitung, untuk suatu kumpulan titik terurut (P1, P2, ..., Pn) yang membentuk poligon tak-memotong, berapa luas poligon tersebut. Perhitungan dilakukan dengan menjumlahkan determinan (P1 P2), (P2 P3), ..., (Pn-1 Pn), (Pn P1) kemudian mengambil nilai mutlaknya dibagi dua.
23.	find_ear	Fungsi find_ear mencari sebuah segitiga anggota sebuah poligon, dimana jika segitiga tersebut dipotong dari poligon tak-memotong, maka hasilnya merupakan poligon tak-memotong juga. Algoritma memeriksa untuk setiap kelompok titik (Pi, Pi+1, Pi+2) dari set titik terurut yang membentuk poligon tak-memotong, apakah luas segitiga tersebut ditambah poligon hasil potongan sama dengan luas poligon awal, serta apakah poligon hasil potongan memiliki dua atau lebih sisi yang saling memotong.
24.	poligonPointsToTriangles	Fungsi poligonPointsToTriangles menerima input kumpulan titik terurut (P1, P2, ..., Pn) yang membentuk poligon tak-memotong dan mengembalikan kumpulan titik terurut

		(A1, A2, A3, B1, B2, B3, ...) dimana setiap tiga titik membentuk segitiga dan gabungan semua segitiga membentuk poligon awal. Algoritma yang digunakan untuk melakukan triangulasi adalah ear-clipping algorithm yang mencari ear dari poligon menggunakan fungsi find_ear lalu memotong segitiga tersebut dari poligon dan mengulang algoritma pada poligon tersisa hingga habis.
25.	showHelp, closeHelp	Fungsi untuk menampilkan dan menyembunyikan menu help

Berikut ini adalah hasil dari program yang telah kami buat.

File Name:

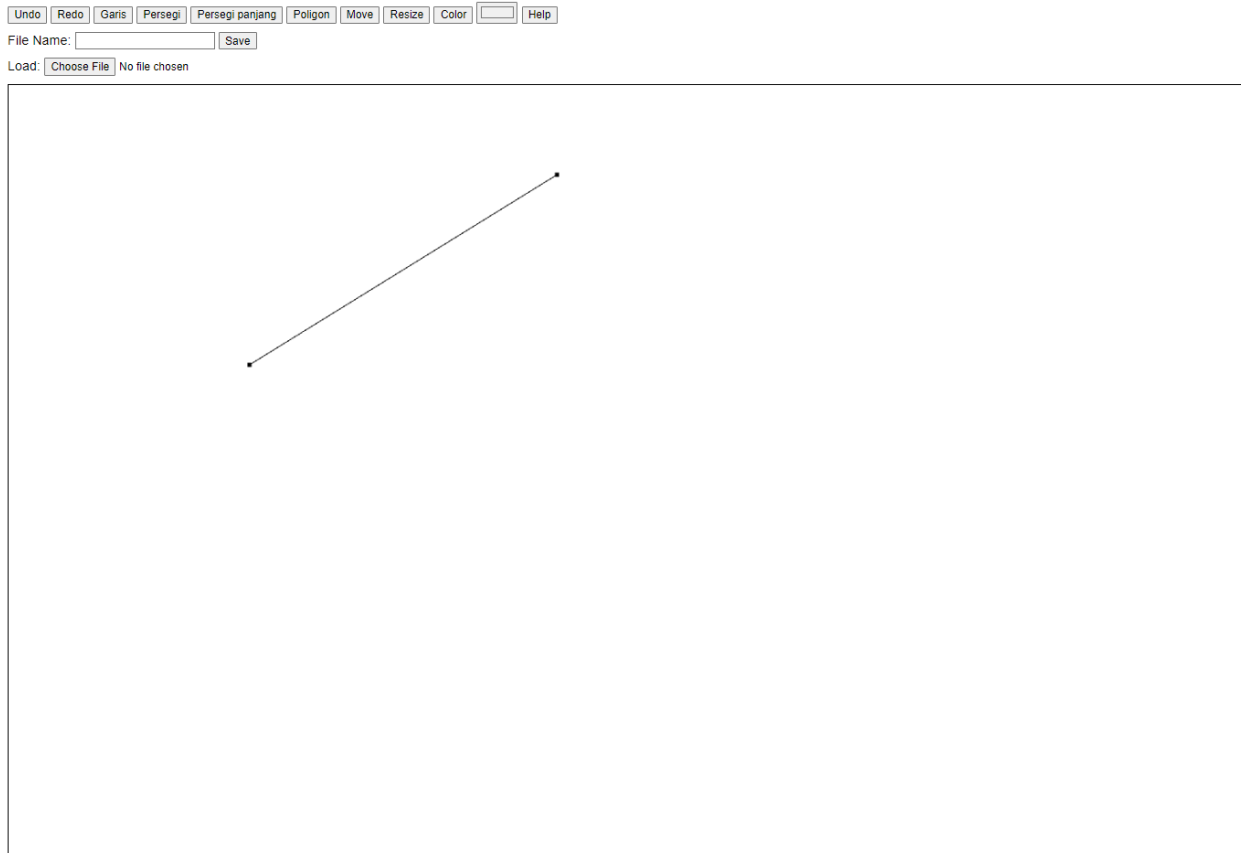
Load: No file chosen



III. Manual/Contoh Fungsionalitas Program

A. Membuat Garis

Fungsionalitas pertama yaitu membuat garis, Anda dapat melakukan ini dengan memilih pilihan **Garis** pada tombol di atas *canvas* kemudian Anda dapat klik pada dua buah tempat di *canvas* untuk membuat garis. Contoh hasilnya adalah sebagai berikut.



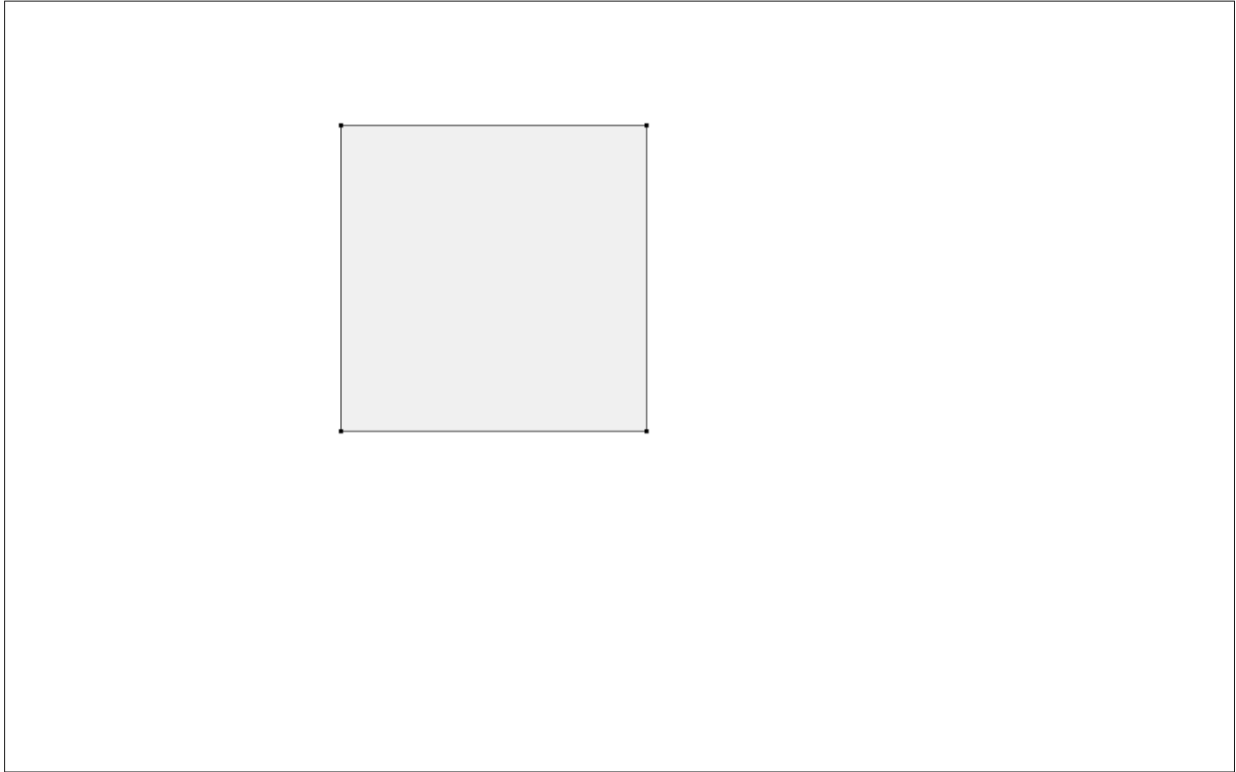
B. Membuat Persegi

Fungsionalitas kedua yaitu membuat persegi, dilakukan dengan memilih pilihan **Persegi** pada tombol di atas *canvas*. Anda dapat klik untuk membuat persegi lalu ditekan seraya melakukan drag untuk mengatur ukuran dan posisi persegi yang diinginkan. Perlu diperhatikan bahwa persegi akan menghasilkan sisi yang sama panjang. Contoh hasilnya adalah sebagai berikut.

Undo Redo Garis Persegi Persegi panjang Poligon Move Resize Color Help

File Name: Save

Load: No file chosen



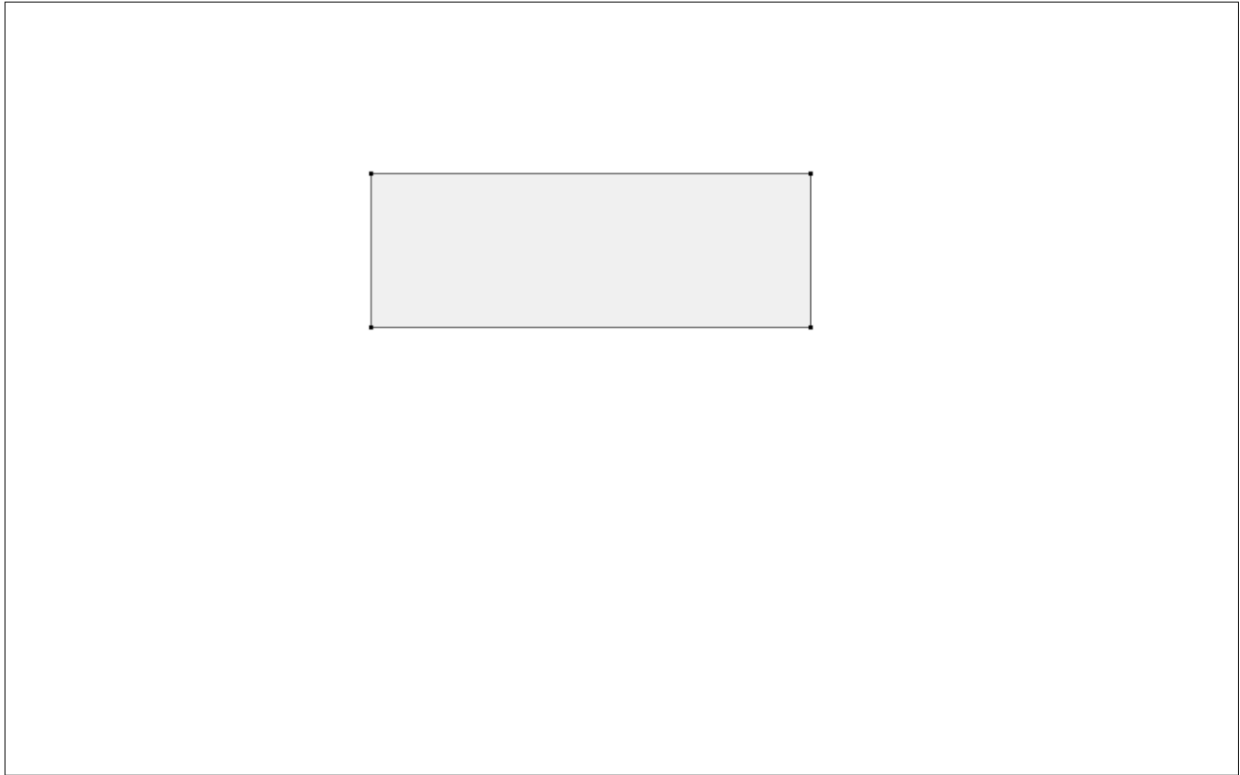
C. Membuat Persegi Panjang

Fungsionalitas ketiga yaitu membuat persegi panjang, dilakukan dengan memilih pilihan **Persegi Panjang** pada tombol di atas *canvas*. Anda dapat klik untuk membuat persegi panjang lalu ditekan seraya melakukan drag untuk mengatur ukuran dan posisi persegi panjang yang diinginkan. Contoh hasilnya adalah sebagai berikut.

Undo Redo Garis Persegi Persegi panjang Poligon Move Resize Color Help

File Name: Save

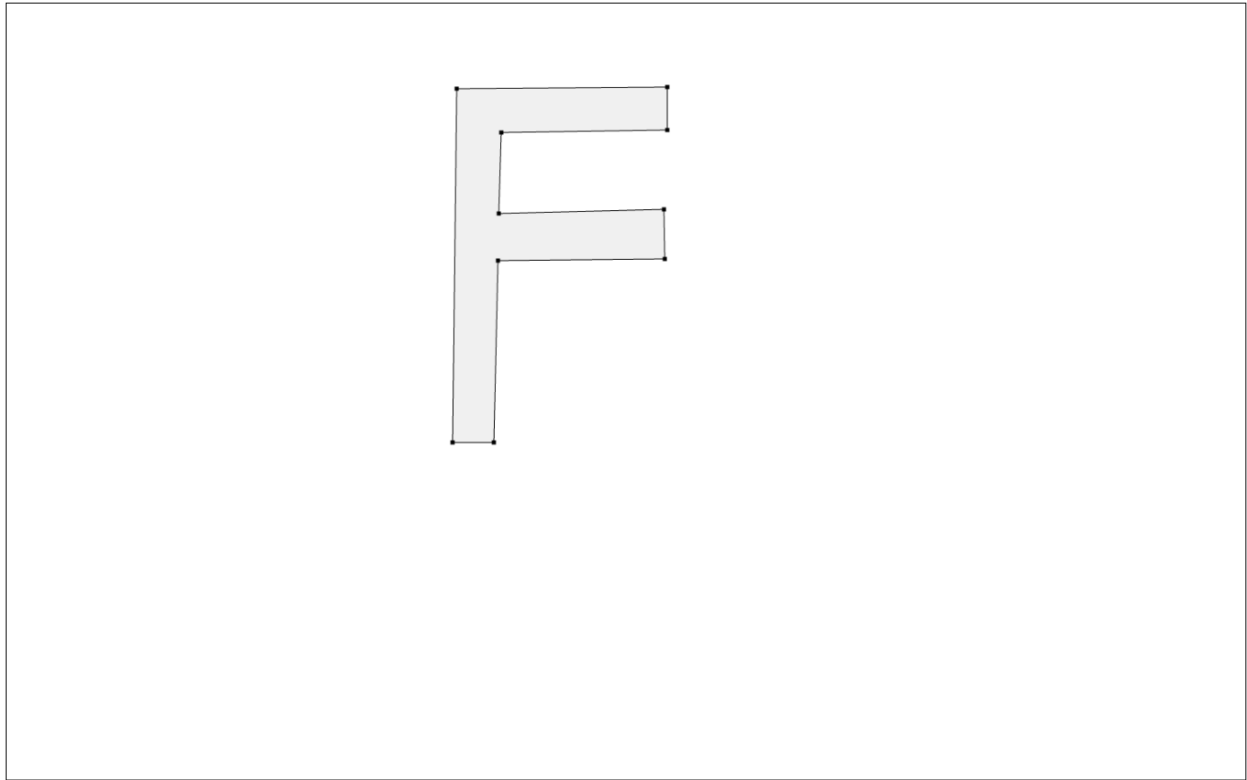
Load: No file chosen



D. Membuat Poligon

Fungsionalitas keempat yaitu membuat poligon, dilakukan dengan memilih pilihan **Poligon** pada tombol di atas *canvas*. Anda dapat klik beberapa kali di tempat yang berbeda pada *canvas* untuk membuat suatu poligon. Perlu diperhatikan bahwa tidak boleh ada garis yang memotong ketika membuat sebuah poligon. Contoh hasilnya adalah sebagai berikut.

Undo Redo Garis Persegi Persegi panjang Poligon Move Resize Color Help
File Name: Save
Load: No file chosen



E. Mengubah Warna Model

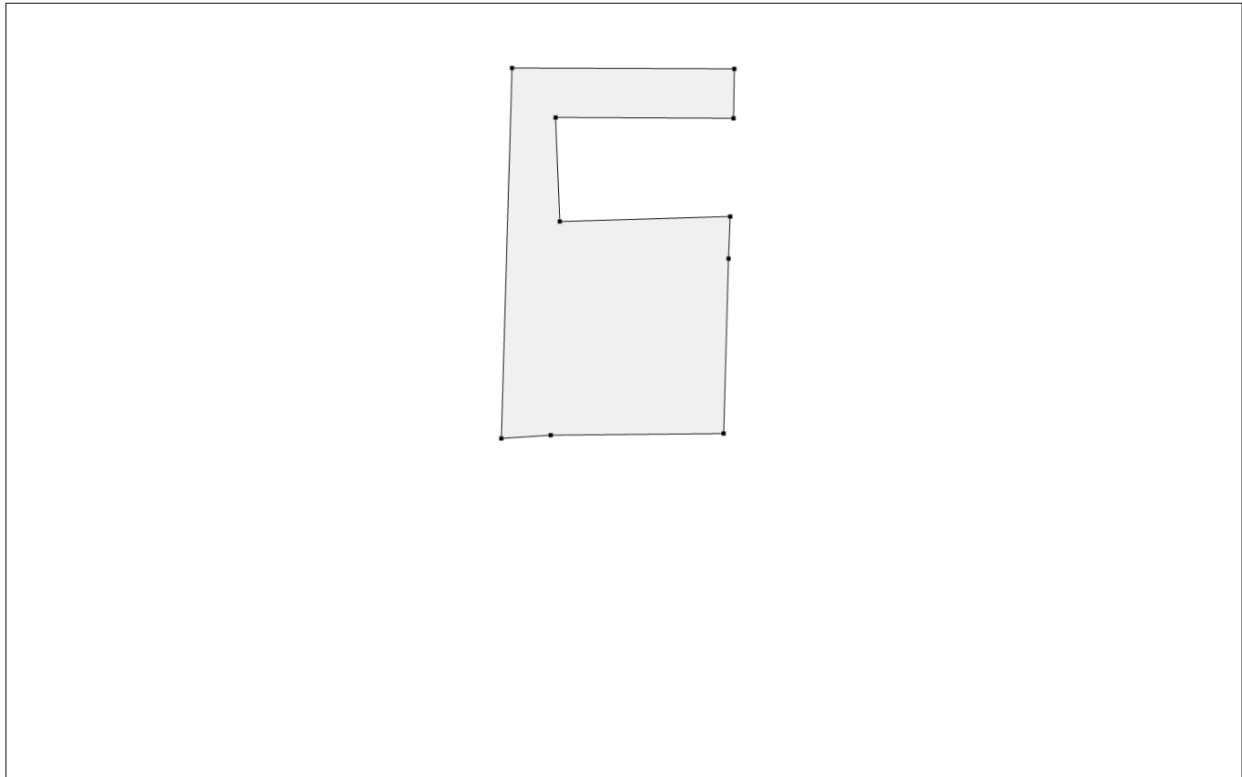
Fungsionalitas kelima yaitu mengubah warna model, dilakukan dengan memilih pilihan **Color** pada tombol di atas *canvas*. Sebelumnya, Anda dapat memilih warna yang diinginkan dengan mengklik tombol **warna** yang berada sebelah kanan tombol **Color**. Setelah memilih tombol **Color** anda dapat mengklik pada model yang ingin diubah warnanya. Contoh hasilnya adalah sebagai berikut.



F. Menggeser Titik Simpul

Fungsionalitas keenam yaitu menggeser titik simpul dapat dilakukan untuk menggeser titik simpul garis atau titik simpul poligon. Fungsionalitas ini dilakukan dengan memilih pilihan **move**, menekan titik yang akan digeser, lalu menggeser titik dengan menggerakkan *mouse* ke posisi yang baru, terakhir melepaskan *mouse* untuk memfinalisasi letak titik baru. Contohnya pada gambar di bawah mula mula poligon berbentuk huruf 'F' kemudian ketika menggeser titik pada huruf 'F' hasilnya adalah seperti gambar di bawah ini.

Undo Redo Garis Persegi Persegi panjang Poligon Move Resize Color Help
File Name: Save
Load: No file chosen



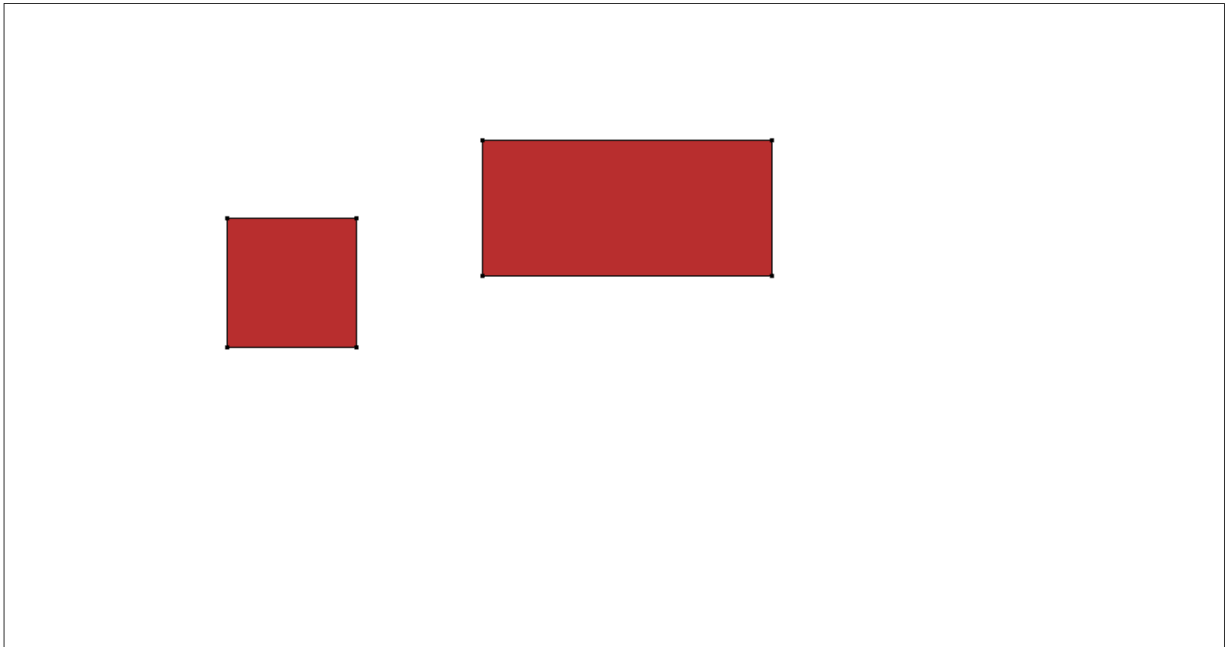
G. Mengubah Panjang Sisi Persegi atau Persegi Panjang

Fungsionalitas ketujuh yaitu mengubah panjang sisi persegi atau persegi panjang dilakukan dengan memilih pilihan **resize**, menekan titik sudut pada persegi atau persegi panjang yang akan diubah ukurannya, lalu menggeser titik tersebut dengan menggerakkan *mouse* ke posisi yang baru, terakhir lepaskan *mouse* untuk memfinalisasi ukuran sisi yang baru.

Undo Redo Garis Persegi Persegi panjang Poligon Move Resize Color  Help

File Name: Save

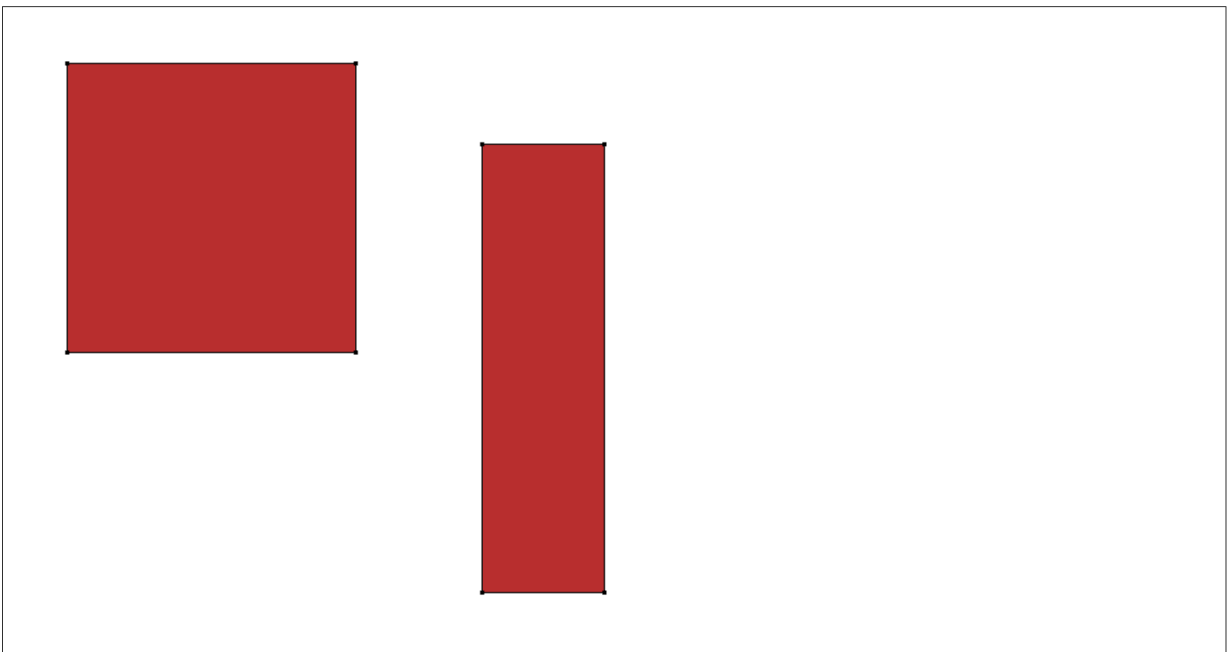
Load: No file chosen



Undo Redo Garis Persegi Persegi panjang Poligon Move Resize Color  Help

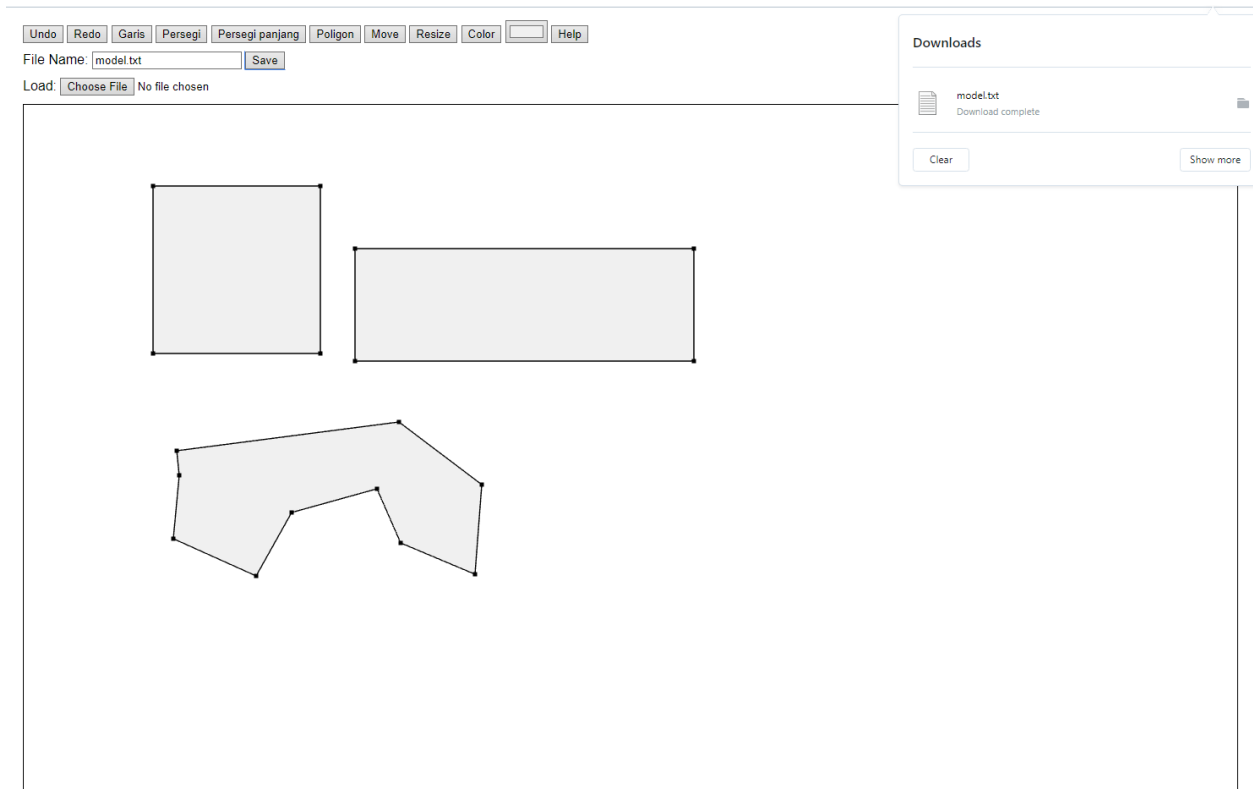
File Name: Save

Load: No file chosen



H. Menyimpan Model

Fungsionalitas kedelapan yaitu menyimpan model dapat dilakukan dengan mengisi nama file pada bagian **File Name** memilih pilihan **Save**. Model yang telah dibuat kemudian akan didownload oleh browser.



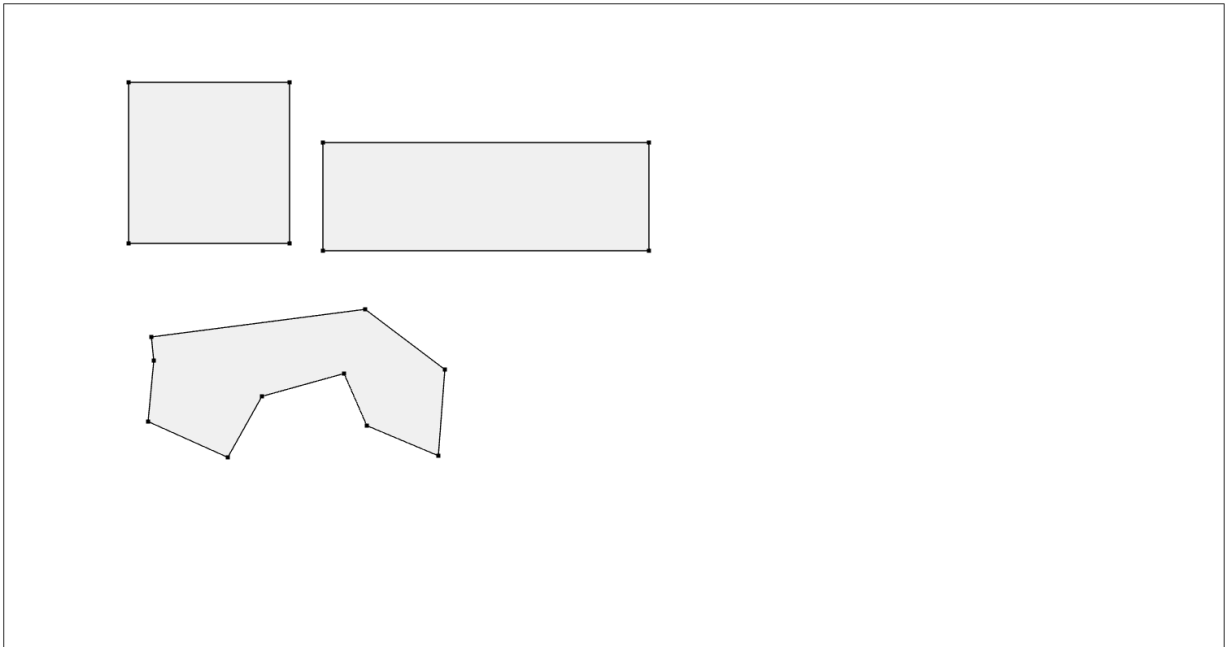
I. Membaca Model

Fungsionalitas kesembilan yaitu membaca model dapat dilakukan dengan menekan tombol **Choose File** dan memilih file model yang akan dibaca. Model yang dipilih kemudian akan ditampilkan dalam canvas.

Undo Redo Garis Persegi Persegi panjang Poligon Move Resize Color Help

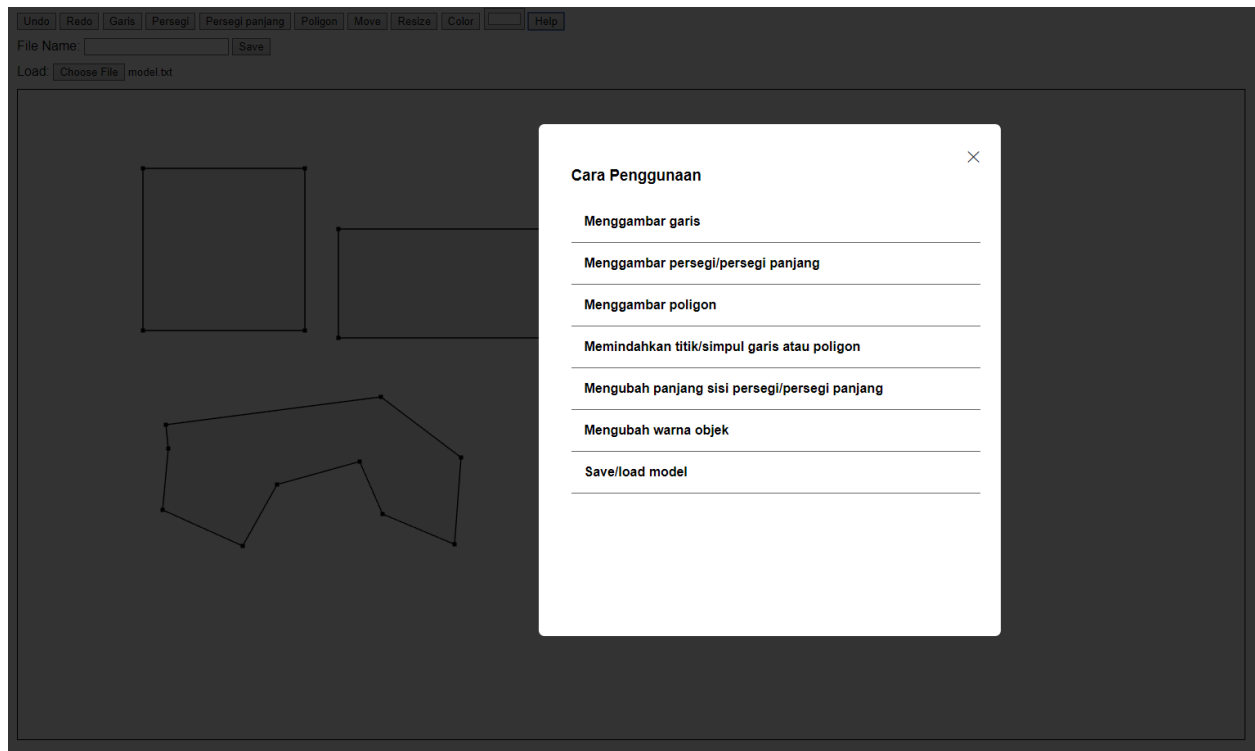
File Name: Save

Load: model.txt



J. Menampilkan Menu Help

Fungsionalitas kesepuluh yaitu menampilkan menu help dapat dilakukan dengan menekan tombol **Help**. Untuk menutup menu help dapat dilakukan dengan melakukan klik pada tombol silang



IV. Pembagian Tugas

Nama	NIM	Pekerjaan
Nabil Nabighah	13519168	Dokumen, Membuat persegi, Membuat move titik simpul pada garis dan poligon.
Ahmad Saladin	13519187	Dokumen, Membuat persegi panjang, Mengubah panjang sisi persegi atau persegi panjang, Menyimpan dan membaca model.
Muhammad Rifat Abiwardani	13519205	Dokumen, Membuat garis, Membuat poligon, Membuat program dari shader objek, Mengubah warna, Menampilkan panduan, Undo dan redo.